Memory for Continually Changing Information: A Task Analysis and Model of the Keeping Track Task

Wolfgang Schoppek (wschoppe@gmu.edu)
Human Factors and Applied Cognition Program
George-Mason University
Fairfax, VA 22030-4444, USA

Abstract

Keeping track of continually changing information has been investigated since Yntema & Mueser's (1960) seminal work. The fact that types of mappings between objects and values and of memory load affect performance are well established, but have never been integrated in a theory. As a step toward such a theory, this paper describes a mathematical model that combines a task analysis with a set of assumptions derived from the ACT-R theory about the dynamics of memory traces. The model's remarkable reproduction of data published by Venturino (1997) demonstrates that standard memory concepts are sufficient to explain the results related to this paradigm. The model yields a clear implication about what causes interference and helps specify open questions.

In many areas of supervisory control, operators have to keep track of the changing values of a number of variables. Knowing the current state of a dynamic system is an important component of situational awareness (Endsley, 1995). For example, a pilot flying a modern automated aircraft needs to know the current altitude, speed, and course of the aircraft, the current settings and modes of the flight management system, just to mention a few of the variables.

In the experimental paradigm for keeping track of continually changing information, introduced by Yntema and Mueser (1960), object-value pairs are presented successively, interrupted by queries about the value associated with a certain object. The most common variables manipulated are the number of objects and the number of attributes from which the values are selected.

In Yntema and Mueser's (1960) experiment, subjects either had to keep track of changing values of many attributes for one object or changing values of the same attribute for many objects. Memory performance was worse in the latter condition. This was attributed to a high degree of interference when only one attribute is used.

Venturino (1997) argued that Yntema and Mueser (1960) confounded attribute similarity and information organization. Figure 1 illustrates how the former factor is defined by the number of attributes, the latter defined by the number of objects. In order to investigate the relative influence of the two factors on memory performance, Venturino (1997) completely crossed these two factors, such that all four possible combinations between high and low attribute similarity and high and low information organization were included. A third factor was memory load. Attribute similarity had a large effect on memory performance, which confirmed Yntema and Mueser's (1960) findings. Information organization also had a significant effect, but this effect was much weaker. As expected, performance declined with memory load.

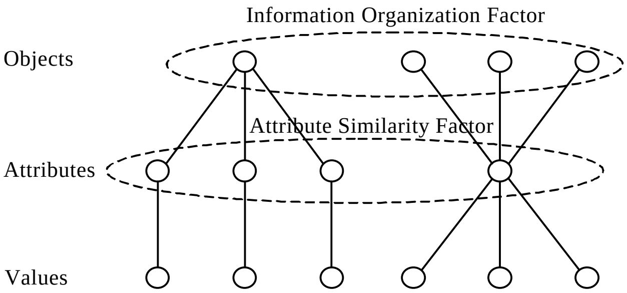

Figure 1: Illustration of the relations between objects, attributes, and values in the paradigm of continually changing information

This same paradigm was used by Hess, Detweiler and Ellis (1999) to prove the superiority of spatially rich displays over displays that show values of different attributes in the same location. Although their research goal was different from Yntema and Mueser's, the basic findings of the paradigm were confirmed in these experiments.

To summarize, the effects of attribute similarity and of memory load are well established. Although the main effects can be explained through the interference that occurs between values of the same attribute, the interactions between attribute similarity and memory load are understood less well. There is no integrative theory that accounts for all the effects. Venturino (1997) interpreted his results as suggesting a distinction between memory capacity for static information and memory capacity for dynamic information, because memory performance in the same-attribute condition was worse than what would be expected in a comparable static memory task.

The goal of this work is to explore if the results about keeping track of dynamically changing information can be explained more parsimoniously with standard assumptions about memory. As a means for this exploration, I developed a mathematical model of the experiment by Venturino (1997). The model combines a task analysis with a set of assumptions about the dynamics of memory traces that are derived from the ACT-R theory (Anderson & Lebiere, 1998). The model may also contribute to an integrated understanding of all the effects related to the paradigm.

In the following sections, I first describe Venturino's experiment in more detail before I present and discuss the model.

Venturino's Experiment

The material used in the experiment consisted of the names of six different fire engines and six different attributes with six values each. Continually changing attribute values were assigned to the fire engines. The task was to memorize these values. After a series of five to seven updates, the subject was asked for the current attribute value of a certain fire engine. For example, in keeping track of the current values of two fire engines, a subject might have to keep track of the number of firefighters for a pumper engine and the location of a tanker engine.

This continual updating is shown with a detailed example in Table 1. Time is represented in discrete steps, where 1 denotes the time of the most recent update, 2 the time step before, and so on. I will refer to these steps as lag, indexed by the variable i.

Table 1: Illustration of the continual updating of values (asterisks indicate an updating event)

lag	stimuli		current value of		
time	fire engine	n fire-fighters	tanker	ladder	pumper
…	…	…	…	…	…
4	tanker	4	*4	…	…
3	ladder	7	4	*7	…
2	tanker	5	*5	7	…
1	pumper	4	5	7	*4
now					

The example shows tanker being updated with the value four at lag 4, ladder being assigned the value seven at lag 3, tanker updated with the value five at lag 2, and pumper being assigned the value four at lag 1. Every fire engine keeps its value until it is updated. These update events are indicated by asterisks in Table 1. The three current values "now" are five firefighters for tanker, seven firefighters for ladder, and four firefighters for pumper. Note that the values differ in "age".

Three independent variables were manipulated in the experiment: number of objects (one vs. many fire engines), attribute similarity (same vs. different attribute), and memory load (two, four or six values to keep track of). The first two factors were varied between subjects; the last factor was varied within subjects. In the many-object/different-attributes condition, unique mappings between objects and attributes were used, such that each of the two, four, or six engines had a value of a different attribute. In the many-objects/same-attribute condition, two, four, or six fire engines had multiple values of the same attribute. In the one-object/different-attributes condition, one engine had values of two, four, or six attributes. In the one-object/same-attribute condition, one fire engine had a value of one attribute. In order to manipulate memory load in this condition, subjects had to memorize the history of the last two, four, or six values. Despite the different mappings, the same number of values had to be remembered in each memory load condition.

Each block began with an initialization of values, followed by 75 to 105 updates, presented at a rate of one update each seven seconds. The updates were randomly interrupted by 15 queries. There were100 subjects total, randomly assigned to one of the four conditions. In a first session, subjects studied the experimental material and, after a few practice trials, worked on the block with memory load 2. Two days later, the blocks with memory load 4 and 6 were administered.

Performance was measured as the proportion of correct answers. The outlined markers of Figure 3 illustrate the main results. All three independent variables had significant main effects on performance, but they were differently strong. Attribute similarity accounted for 15% of the variance, information organization (number of objects) for only 1%. The main effects were qualified by a significant three-way interaction of all factors. Separate analyses revealed significant interactions between attribute similarity and memory load in both object conditions: Memory load affects performance much more when the same attribute is used than when different attributes are used.

In the same-attribute condition, there was a significant interaction between memory load and number of objects: In the many-object condition performance decreased more sharply as memory load increased than in the one-object condition. In the different-attribute condition, the number of objects had no significant effect on performance.

An error analysis revealed that 44% of the errors were previous state errors, i.e. a subject responded with the previous value of an attribute rather than its current value. Interestingly, subjects responded significantly faster (M = 4.58 s) when making a previous state error than when making any other type of error (M = 5.30 s).

Model

In this section, a model will be described that is able to reproduce the results of Venturino's experiment. The predictions of the model are not derived from simulation, but from a mathematical combination of the probabilistic

Table 2: Task analysis of the keeping track task

lag i	fire-engine	value v_i	v_i current now?	p (v_4 current after lag i)	p (v_3 current after lag i)	p (v_i current now) = q_i
4	tanker	4	no	1	…	0.75^3
3	ladder	7	yes	0.75	1	0.75^2
2	tanker	5	yes	0.75^2	0.75	0.75
1	pumper	4	yes	0.75^3	0.75^2	1
now						

structure of the material and basic assumptions about the dynamics of memory elements. The psychological assumptions originate from the ACT-R theory (Anderson & Lebiere, 1998).

Suppose that each update event is stored as a unique memory trace. The probability that this trace contributes to a correct answer equals the probability that the trace represents a current value times the probability that it is retrieved from memory. The first factor is given by the task analysis described below, the second factor is derived from a cognitive model. Summing up the probabilities of contributing to a correct answer for all memory traces gives an estimate of the number of correct answers for all possible probes.

Task Analysis

The first component of the model is an analysis of the probabilistic structure of the material used in the experiment. This task analysis allows us to determine the probability that a value is current as a function of the update time and the memory load condition.

Table 2 is built on the example given in Table 1 and contains information that is relevant to understanding the task analysis. Time is again indicated by lag. The values that were presented at each time step are referred to as v_i. Column 4 contains the "currency" of the respective values v_i at present time (now), i.e. immediately after lag 1. The values v_1, v_2, and v_3 are still current, but v_4 is not, because it was overwritten with v_2. Column 5 shows the probability of v_4 being current at the end of each time step. At the end of lag 4, v_4 is current (probability equals 1.0), because it has just been updated. At lag 3, one of the four vehicles is randomly chosen for an update. Thus, the probability of v_4 being updated at lag 3 is 0.25. Put another way, the probability of v_4 being current at the end of lag 3 is 1-0.25 = 0.75. The same considerations hold for the following steps.

Because the updates are independent events, the probabilities for each time step must be multiplied to obtain the overall probability that a value is still current. Thus, the probability of v_4 being current after lag 1 ("now") is 0.75^3. Column 6 exemplifies that for the update of "ladder" at lag 3. The last column of Table 2 contains the resulting probabilities of being still current for v_1 through v_4. Equation 1 is the generalized form of the probability q_i of value i still being current.

$$q_i = p_s^{i-1} \tag{1}$$

In Equation 1, p_s is the probability of not being updated in the following step. This variable depends on the memory load n_c (i.e. number of current values given by the number of vehicles and/or attributes), according to Equation 2.

$$p_s = 1 - 1/n_c \tag{2}$$

Applying Equations 1 and 2 to Venturino's experimental materials results in the probabilities depicted in Figure 2. Each memory load condition results in one curve. Memory load condition 6 involves six current values, distinguished by the type of vehicle, the attribute, or a unique mapping between vehicle and attribute. Similarly, memory load conditions 4 and 2 involve four and two values, respectively. It is obvious that the probabilities of being current diminish much faster the fewer current values there are, because the probability for each value being updated is higher when there are fewer dimensions (attributes and/or objects).

The task analysis also reveals that the probabilistic structure of the one-object/same-attribute condition deviates considerably from this scheme. Because in this condition, the last two, four, or six values of the same attribute have to be remembered for only one object, the probabilities of these values being current are one, the probabilities of all other values are zero. This different structure was entered at the appropriate places in order to calculate the model's prediction.

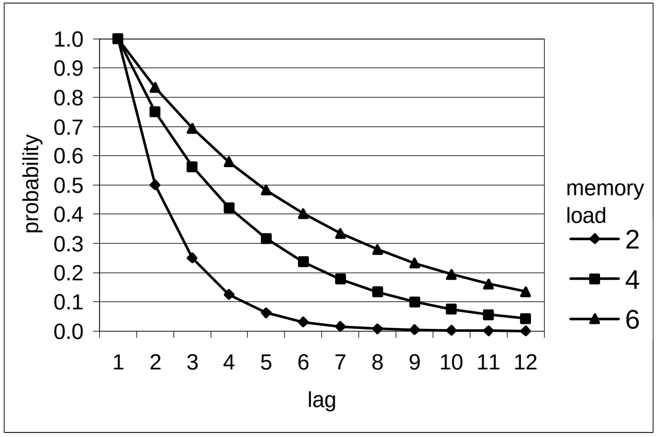

Figure 2: Probabilities q_i that a value that was updated in a certain time step is still current.

Cognitive Model

The second component of the model is a set of assumptions about the dynamics of the memory representations that are formed from the update events. The first assumption is that for each update event a new memory element is created which represents the information given in the update. The second assumption is that each element is rehearsed a number of times, thus being strengthened. The remaining assumptions are part of the ACT-R theory.

According to the rational analysis basis of ACT-R, the activation of a declarative memory element reflects the probability that the element is needed in the current context and determines its retrieval. The two additive components of activation are baselevel activation and net activation. The former reflects the baserate probability, the latter the conditional probability given the current context. In this application, current context means the cues that are active and enhance retrieval of the correct memory element. Since this model makes no specific assumptions about cues, we can focus on baselevel activation.

The baselevel activation of an element is defined as the log odds that the element is needed. The odds are calculated with Equation 3, where *n* is the number of times the element has been needed, and *L* is the lifetime of the element[1]. Lifetime is the time that has passed since the creation of the element. The more frequently a memory element has been needed in its lifetime, the higher is its baselevel activation. If an element is not needed for some time, its baselevel activation decays. These changes of baselevels depending on use and time are referred to as *baselevel learning*.

$$odds = \frac{2n}{\sqrt{L}} \qquad (3)$$

As mentioned earlier, I assume that a new memory element is created for each updating event and that this element is rehearsed a number of times after its creation. Each single rehearsal involves a retrieval of the element, which increases the respective *n*. The number of rehearsals is a free parameter of the model. The lifetime *L* is determined by the lag at which the element was created and the duration of each step (which was seven seconds in Venturino's experiment).

Odds can be transformed into probabilities using the definition *odds* = *p*/(1-*p*). This gives us Equation 3a.

$$p = odds/(odds+1) \qquad (3a)$$

With this equation, the probability of retrieval *p* can be predicted for each memory element that was created to represent an update event.

This probability is assumed to be degraded in the same-attribute conditions where interference is expected, depending on the number of competing memory elements. Assuming that only the elements that represent current values of the same attribute are competing, the respective numbers n_c are two, four, and six. Note that in the different-attribute conditions there is only one current value of each attribute, so no interference is expected there.

I assume further that the interference effect is "buffered" by a constant *c*, which is the second free parameter of the model. Equation 4 shows the degrading function. To ensure that the degraded probability value ranges between 0 and 1, *c* may vary between 0 and 0.5.

$$p' = \left\{ \begin{array}{c} p \mid \text{condition = different - attributes} \\ p \cdot (c + 1/n_c) \mid \text{condition = same - attribute} \end{array} \right\} (4)$$

It is important to realize that the cognitive component of the model makes no assumptions about the influence of the information organization factor. This can be justified by the result that this factor accounted for only 1% of the variance in the experiment. Nevertheless, the predictions for the one-object conditions are slightly different from those for the many-object conditions, because of the different probability structure of the one-object/same-attribute condition.

Equation 5 describes how the prediction of the model is obtained by summing up for each time step the probability that its value will lead to a correct answer and dividing the sum by the number of current values (i.e. memory load). q_i is the probability that the value of step *i* is still current, p_i' is the probability that the memory element representing that value is retrieved, and n_c is the number of current values.

$$P = \frac{\sum_{i=1}^{s} q_i \cdot p_i'}{n_c} \qquad (5)$$

Summing up the probabilities of all memory traces gives a generalized estimate of their potential to answer all possible probes. The prediction of the model should be the expected proportion of correct answers. Therefore, the sum must be divided by the number of current values, because, depending on memory load, all traces contain two, four, or six traces that represent current values.

The two free parameters of the model, number of rehearsals *n* (Equation 3) and *c* (Equation 4), were estimated to optimize the fit to the data. The resulting values were *n* = 12 rehearsals and *c* = 0.5. With these values, the prediction of the model matched the data with an R^2 of 0.89 and a root-mean-square deviation (RMS) of 0.07.

[1] Equation 3 is an approximation of the original ACT-R equation. The approximation includes the default value 0.5 of the "baselevel-learning" parameter. The similarity between the time functions of Equations 1 and 3 illustrates the ACT-R notion that memory processes reflect the probabilistic structure of the environment.

Although an R^2 of 0.89 might not seem very high, one has to take into account that twelve degrees of freedom were predicted by adjusting only two parameters. For the many-objects conditions alone, the R^2 is 0.97 and the RMS is 0.04.

Note that the task analysis contributes to the prediction only in combination with the memory assumptions. Since Σq_i equals n_c (cf. Equations 1 and 2), a constant probability of retrieval p' would simplify the numerator of Equation 5 to $n_c \cdot p'$, and Equation 5 would yield the constant p'. The variation of probabilities of being current, q_i , would be completely neutralized by a constant probability of retrieval, p', and no differences would be predicted.

If only the assumption about baselevel learning would be omitted, Equation 4, which models the interference effect, would still create variations in p' . I tried to fit the data without the calculation of retrieval probabilities as a function of time (i.e. without baselevel learning), using a single value for the probability of retrieval p. This value was estimated as $p = 0.85$. The resulting values of $R^2 =$ 0.77 and RMS = 0.08 show that the interference assumption alone accounts for a fair amount of variability, but the prediction is clearly improved by the assumption about baselevel learning.

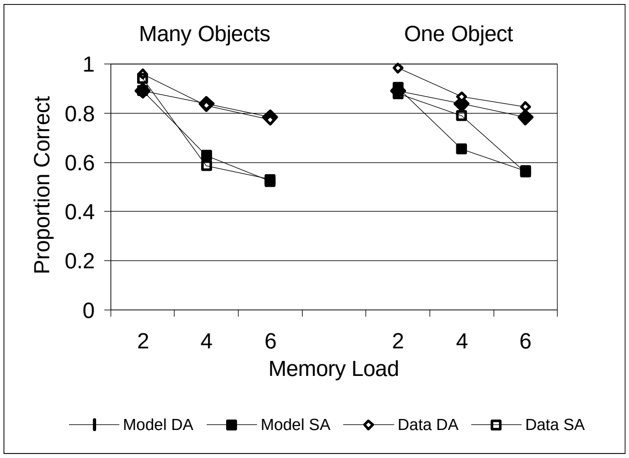

Figure 3: Mean proportions of correct answers from Venturino (1997) and the model (DA: different attributes, SA: same attribute)

Discussion

It is remarkable that a model that combines a task analysis with a small set of basic assumptions about the dynamics of memory elements can reproduce the data so well. This demonstrates that there is no reason to distinguish between memory capacity for static information and memory capacity for dynamic information, as it was suggested by Venturino (1997). The model implies a simple rehearsal strategy in which only the most recent value is rehearsed about twelve times. This number is slightly higher than the number of rehearsals that were needed to encode the instruction in a model of serial attention by Altmann (2000). Because the present model does not include activation spread by cues, which would also increase the probabilities of retrieval, this number of rehearsals is probably overestimated.

The simplicity of the rehearsal strategy was not assumed for sake of parsimony, but is actually functional. If more than the most recent value would be rehearsed, this would strengthen older memory traces to a degree that new traces could hardly compete with the older ones, thus preventing the system from retrieving newer traces which are more likely to represent current values. This prediction of the model should be tested in future research.

Although the model is successful with standard assumptions about memory, there is one feature that points in a similar direction as Venturino's (1997) speculation about different types of memory capacity. The parameter c in Equation 4 and its estimated value of 0.5 establish a threshold of two memory elements up to which no interference occurs. This raises the question if there might be a preferential type of representation for a very small number of elements. Such an assumption, implemented in a simulation model, would remedy the model's underestimation of performance in the lowest memory load conditions. ACT-R provides opportunities to model such a preferential representation, for example if one assumes that one or two of the most recent values are always elements of the focus of attention.

Another interesting question that can be stated more precisely thanks to the model is what interferes with the correct answer. The present model assumes that only the *current* values that share the same attribute interfere with each other, resulting in no interference in the conditions with different attributes. The small memory load effect in these conditions is due to the increasing mean "age" of the memory representations with higher memory load. Also in the same-attribute conditions, the interference factor (Equation 4) depends on the number of *current* values.

This assumption, although critical for the predictions and supported by the data, can be questioned. It might be more plausible to assume that not only the current values of an attribute compete, but all of them. Interestingly, this assumption predicts more interference for lower memory loads in the different-attribute conditions. Suppose there are twelve memory elements representing the twelve most recent values, some of them current, some not. Under memory load 2, there are two different attributes, thus on average six of the elements share the same attribute. Under memory load 4, three elements, and under memory load 6, two elements share the same attribute. Thus, the lower the memory load, the more elements of the same attribute compete with each other, producing higher interference - a pattern that is contradicted by the data.

All these observations converge at the question of what happens with the memory elements that represent outdated values. The decay of baselevel activation certainly contributes to the diminishing interference potential of outdated memory elements, but the decay guarantees this effect only if no noise is assumed. If one assumes some noise, which seems to be realistic, much more interference would be expected than predicted by the present model and found in the data. I have started to investigate this problem using a rather process oriented, symbolic type of modeling. It will be interesting to see if additional processes such as active inhibition have to be assumed to explain the rather low interference effects.

Another advantage of symbolic modeling is that it demands more details about cues. In the present model, it was implicitly assumed that only one strong cue is in effect. It is the attribute in the different attribute condition. In this condition, only one value of each attribute is a current value. This value is always the most recent - and thus the most active value of that attribute. Therefore, using the attribute as a constraint and retrieving the most active memory element delivers the correct answer.

The reason why the attribute is assumed to be the only strong cue is that the relation between an attribute and its values is the only one that stays constant throughout the experiment. In their Experiment 4, Hess et al. (1999) established a constant relation between a spatial cue and attribute values in a many-objects/same-attribute condition. This cue was strong enough to abolish the interference effect that is usually observed in that condition.

The objects one the other hand are much less potent cues, because the relation between objects and attribute values varies. This is probably the reason why the information organization factor (which is operationalized through the number of objects) exerts so little influence. The model even justifies to doubt if there is a real effect at all, because the difference between the one-object and many object conditions is partially explained by the different probability structure of the material in the one-object/same-attribute condition. One data point that contributes much to the difference is the performance in memory load 4 of that same condition where the model's predictions deviate most highly from the data. A replication would be necessary to find out if this deviation is rather due to noise in the data or to inappropriate assumptions of the model. In such a study, the probability distribution of the one-object/same-attribute condition should be approximated to the distributions of the other conditions in order to draw clearer conclusions about information organization.

Conclusions

The model has demonstrated clearly that a task analysis combined with a small set of assumptions about the dynamics of memory traces is sufficient to reproduce the basic results related to the keeping track paradigm. No distinction between memory capacities for static and for dynamic information is needed. The model implies that interference occurs between representations of current values. Hence, an issue of future research should be to investigate what happens with the representations of outdated values. As to the factor information organization, it has been shown that the effect of this factor is partially due to the deviating probability structure of one of the conditions. To clarify the influence of information organization, the probability structures should be assimilated in future studies by means of the presented task analysis.

Acknowledgments

I wish to thank Mike Venturino for providing me with detailed information about his experiment. I am grateful to my colleagues at the Human Factors and Applied Cognition Program, especially to Deborah Boehm-Davis and Erik Altmann for their comments on this paper. Thanks also to Wayne Gray who encouraged me to model this paradigm. This research has been supported by grants NAG 2-1289 from the NASA and 99-G-010 from the FAA.

References

Altmann, E. (2000). The anatomy of serial attention: An integrated model of set shifting and maintenance. *Proceedings of the Third International Conference on Cognitive Modeling*, Groningen, The Netherlands.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, *37*, 32-64.

Hess, S. M., Detweiler, M. C., & Ellis, R. D. (1999). The utility of display space in keeping track of rapidly changing information. *Human Factors*, *41*, 257-281.

Venturino, M. (1997). Interference and information organization in keeping track of continually changing information. *Human Factors*, *39*, 532-539.

Yntema, D. B., & Mueser, G. E. (1960). Remembering the present states of a number of variables. *Journal of Experimental Psychology*, *60*, 18-22.